

Project Management in Software Engineering

(Cmpe 451 – Fall 2020)

Instructor	Suzan Uskudarli (suzan.uskudarli@boun.edu.tr)	
Assistants	Meriç Turan (meric.turan@boun.edu.tr) Alper Ahmetoğlu (alper.ahmetoglu@boun.edu.tr) Uğur Acar (ugur.acar@boun.edu.tr)	
Lecture	Section 1 Tuesday 14:00-14:50 Section 2 Tuesday 13:00-13:50	Zoom (from Moodle)
Lab	Section 1 & 2 Tuesday 15:00-16:50	Zoom (from Moodle)

Overview

This course focuses on the successful application of the practices for managing the lifecycle of the development of a software product. It is about applying the fundamental principles gained during Cmpe352. Students will work as a **team** on the entire lifecycle of developing a successful software product. It is a hands-on course including weekly meetings and periodic milestone presentations to track the progress and success of the software project.

Due to COVID related developments, this syllabus may be subject to change if needed.

Course Objectives

1. To experience the application of software development techniques and methodologies throughout various stages of the software development lifecycle including planning, requirements gathering, design, implementation, testing, deployment, and delivery.
2. To acquire team-based software development skills such as communication, planning and time-management.
3. To gain communication skills required to interact with customers and the project team members.
4. To develop skills in planning and managing software development projects.
5. To gain the ability to use software development tools, techniques, environments and methodologies for effective and productive software development.

6. To apply various Computer Engineering knowledge and algorithms, as learnt in earlier courses, to software development.

Reminder of Fundamentals of Software Engineering (Cmpe352) Topics

- Introduction - Software Engineering, Software
- Project and Processes Survey
- How is software developed? Requirements Elicitation.
- Iterative Development Methods – Agile Methodology
- Communication - Team, Customer, Documentation
- Verification and validation — User scenarios –
- User Testing User scenarios, User Test Document
- Functional modeling: Use Cases and Activity –
- Structural modeling: System classes
- Behavioral modeling – Sequence Diagrams – User
- Interface Design Use Cases, Action Diagrams, Class diagrams
- Code version management and Code Review
- Testing, Unit Tests, User Tests, and Debugging
- Project Planning
- Configuration Management
- Lessons Learned

Communication

The official & logistic communication related to this course will be carried out with the following platforms:

Piazza for private and collective communication

<https://piazza.com/boun.edu.tr/fall2020/cmpe451> (active when class starts)

Moodle grades and info about live sessions. (active when class starts)

Zoom: weekly lectures and labs (active for specific sessions)

GitHub: Project related material will be communicated on your GitHub repo. All issue management and documentation must be on GitHub. (already active)

Note that it is your responsibility to keep up with the messages, notifications, and content shared here.

Class Structure

Each student will have been assigned to a software project team that is expected to implement and deploy the project by the end of this semester. All project documentation, issues, and code of the project are maintained and tracked on a code repository (GitHub). Each week we will be discussing software development issues in the context of your team projects. We will reflect and brainstorm about the issues that emerge during your software development process.

Each team member is expected to contribute to all aspects of the project throughout the entire project lifecycle. It is essential to make significant contributions (correctly functioning code that is included in the final solution) to the code base of your team solution.

There will be 3 main milestones (1st, 2nd, and final) that include presentations. These are very important checkpoints! Each team will have further milestones (defined by the team itself) to track the progress. These milestones will be part of the discussions during the weekly meetings.

Weekly meeting goals

Software development is a continuous activity requiring constant attention to the work at hand, time management (planning & tracking), communication, and effective application of software development methods and use of tools.

Each week we will be visiting issues related to your team projects. Your team will make many decisions regarding the implementation of your project that will shape our interactions. Every week you are expected to articulate the development of your project in terms of your personal contribution as well as the overall project status. You will be planning and tracking your project with the tools that support the software development process. In specific, you (any team member) will be expected to answer what the current status is, what work was done in the last week, the differences (if any) between the planned and actual status, and any mid-course actions that you plan to take to remedy any shortcomings. We will ask you to show your GitHub repository which should be kept up-to-date and reflect the current status.

This course is very much about developing software as a **team**, which requires being present, communicating, and fulfilling your responsibilities. Towards this end you will be responsible for planning and tracking your activities so that you may deliver a successful software product. Do not forget that your actions have an impact on all your teammates, which makes it very important to *manage* your time so as to complete your tasks in a timely and correct manner. Needless to say, effective communication is very important.

Assessment and Grading

You will have three main deliverables during this semester, each of which will be the culmination of ongoing work:

- Milestone 1:** the revised requirements, design, and initial prototype
Milestone 2: detailed prototype
Final Milestone: final product deliverable

For these milestones, you will prepare a report as a team deliverable and will make a group presentation. They serve as very important events to assess the desired progress of the project (both for you as well as your customer). All prototypes during milestone presentations must be deployed. Due dates and the details of the deliverables will be discussed in class and announced on Piazza. Be cognizant that all milestones reports will include personal effort tracking records.

During this course you are expected to express yourselves clearly in terms of software engineering terminology, techniques, and concepts. This includes requirements, design documents, issue management, and code. You will be evaluated based on communication (oral and written), effective software development practices and use of tools, managing the software development tasks (requirements, design, coding, testings), and with respect to the execution of the development plan of your project. All of these factors will be considering during the evaluation of your personal and team work:

Criteria	Percent
Participation	10%
Milestone 1	20%
Milestone 2	25%
Final Milestone	45%

Academic Integrity and Collaboration

As computer engineers and software developers adhere to professional code of conduct. Please review the ACM Code of Ethics and Professional Conduct at: <https://ethics.acm.org>.

Complex software development is a team effort, which calls for collaboration. Peer learning and collaboration is welcome in team projects. But, taking credit for work that is not performed by yourself is never acceptable. For example, if you copy and paste chunks of code (from sources like Stackoverflow or Github) without understanding it, you're taking credit for others' work. Likewise, if you copy answers on a quiz or exam from your classmates or from external resources (online or otherwise) you are taking credit for others' work. If you are not clear about what is acceptable as collaboration, please ask us.

If we discover that you are not doing your own work, you will fail this course and may be reported for academic disciplinary action.